

We claim:

1. A computer-implemented method for an object-oriented environment comprising:
wrapping a reference to a second object within a second context with a proxy
wrapper, the second context defining at least a second set of arbitrary invariants on a
5 second set of arbitrary objects including the second object;
calling of the second object within the second context by a first object within a first
context via the reference as wrapped in the proxy wrapper, the first context defining at
least a first set of arbitrary invariants on a first set of arbitrary objects including the first
object; and,
10 returning of the second object within the second context to the first object within the
first context via the reference as wrapped in the proxy wrapper.
2. The method of claim 1, further comprising:
wrapping a reference to the first object within the first context with a proxy wrapper;
calling of the first object by the second object via the reference as wrapped in the
15 proxy wrapper; and,
returning of the first object to the second object via the reference as wrapped in the
proxy wrapper.
3. The method of claim 1, further comprising:
calling of a third object that is agile by the first object via an unwrapped direct
20 reference, such that the first context of the first object becomes a context for the third
object for calling of the third object by the first object, such that the third object executes

in the first context of the first object, and such that the agile object is agile in that the agile object has no permanent context; and,

responding by the third object to the first object via the unwrapped direct reference.

4. The method of claim 3, further comprising:

5 calling of the first object by the third object via an unwrapped direct reference; and,
responding by the first object to the third object via the unwrapped direct reference.

5. The method of claim 1, further comprising:

calling of a third object within the first context by the first object via an unwrapped direct reference; and,

10 responding by the third object to the first object via the unwrapped direct reference.

6. A computerized system comprising:

at least one first object within a first context, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the at least one first object; and,

15 at least one second object within a second context, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the at least one second object,

such that the at least one first object communicate with one another directly via direct references, the at least one second object communicate with one another directly via
20 direct references, and any of the at least one first object communicate with any of the at least one second object via indirect references wrapped in proxy wrappers.

7. The system of claim 6, further comprising at least one agile object, such that the at least one agile object are agile in that the at least one agile object have no permanent context.

8. The system of claim 7, wherein as called by any of the at least one first object any of the at least one agile object executes within the first context, such that any of the at least first object communicate with any of the at least one agile object directly.

9. The system of claim 7, wherein as called by any of the at least one second object any of the at least one agile object executes within the second context, such that any of the at least second object communicate with any of the at least one agile object directly.

10. The system of claim 6, wherein a reference to one of the at least one second object is wrapped in a proxy wrapper, and one of the at least one first object calls the one of the at least one second object via the reference as wrapped in the proxy wrapper.

11. The system of claim 6, wherein a reference to one of the at least one first object is wrapped in a proxy wrapper, and one of the at least one second object calls the one of the at least one first object via the reference as wrapped in the proxy wrapper.

12. A computerized system comprising:

at least one first object within a first context, the first context defining at least a first set of arbitrary invariants on a first set of arbitrary objects including the at least one first object, the at least one first object communicating with one another directly via direct

references;

at least one second object within a second context, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the at least one second object, the at least one second object communicating with one another
5 directly via direct references, and any of the at least one first object communicating with any of the at least one second object via indirect references wrapped in proxy wrappers; and,

at least one agile object, such that the at least one agile object are agile in that the at least one agile object have no permanent context, such as called by any of the at least one
10 first object any of the at least one agile object executes within the first context, and such that as called by any of the at least one second object any of the at least one agile object executes within the second context.

13. A machine-readable medium having a computer program stored thereon for execution by a processor to perform a method comprising:

15 wrapping a reference to a second object within a second context with a proxy wrapper, the second context defining at least a second set of arbitrary invariants on a second set of arbitrary objects including the second object;

calling of the second object within the second context by a first object within a first context via the reference as wrapped in the proxy wrapper, the first context defining at
20 least a first set of arbitrary invariants on a first set of arbitrary objects including the first object; and,

returning of the second object within the second context to the first object within the first context via the reference as wrapped in the proxy wrapper.

14. The medium of claim 13, further comprising:

wrapping a reference to the first object within the first context with a proxy wrapper;

calling of the first object by the second object via the reference as wrapped in the proxy wrapper; and,

5 returning of the first object to the second object via the reference as wrapped in the proxy wrapper.

15. The medium of claim 13, further comprising:

calling of a third object that is agile by the first object via an unwrapped direct reference, such that the first context of the first object becomes a context for the third

10 object for calling of the third object by the first object, such that the third object executes in the first context of the first object, and such that the agile object is agile in that the agile object has no permanent context; and,

responding by the third object to the first object via the unwrapped direct reference.

16. The medium of claim 15, further comprising:

15 calling of the first object by the third object via an unwrapped direct reference; and,

responding by the first object to the third object via the unwrapped direct reference.

17. The medium of claim 13, further comprising:

calling of a third object within the first context by the first object via an unwrapped direct reference; and,

20 responding by the third object to the first object via the unwrapped direct reference.